

DOI: 10.2478/ffp-2023-0008

Accuracy of the evaluation of forest areas based on Landsat data using free software

Paulina Zajac, Ewa Dębińska, Kamil Maciuk ✉

AGH University of Science and Technology, Faculty of Mining Surveying and Environmental Engineering, al. Mickiewicza 30, 30-059 Kraków, Poland, e-mail: maciuk@agh.edu.pl

ABSTRACT

Ever-evolving technologies are enabling us to obtain information about the world around us ever more quickly and precisely. This state of affairs contributes to the growing need to store and analyse data. For today's scientists, this is a challenge because it involves analyses on a global scale. This also applies to spatial data, vast amounts of which are made available online. The Google Earth Engine platform is such a place on the web. It is not just a catalogue for browsing, but above all an environment for programming useful applications. Among free software, it is difficult to find one that is not dependent on the parameters of the computer. In the case of Google Earth Engine, the processes programmed by the user are executed on powerful external servers, and the user only gets the finished result, which he can download to his computer and use in further work. The initial chapters introduce the basic concepts and characterise the specifics of working in this environment, taking into account the limitations of the platform. Then, the individual stages of the algorithm developed by the authors are described, trying to explain well the reasons for the use of particular methods and functions. In order to verify the correctness of the obtained results, existing databases on the subject and published research results of other authors were used.

KEY WORDS

GIS, Google Earth Engine, forest, Landsat, satellite data

INTRODUCTION

One of the valuable regions of our country is the Carpathian Mountains (Kozioł and Maciuk 2020). Their forest area should be monitored out of concern for the state of the environment. The Carpathian forests have been studied by many scientists because of their special natural value (Szramka and Adamowicz 2020; Maciuk et al. 2021). Various data sources have been used in the research. Among the oldest are historical military maps, dating back to the 19th century, the second an Austrian

topographic photograph, where forests were depicted as grey surface signatures (Kozak and Kaim 2016). In order to extract information from these, the area of forest complexes has to be vectorised manually. This method is very time-consuming and labour-intensive. Nowadays, in addition to paper maps, there are various data provided by both state institutions (Central Statistical Office, State Forests), private entities and international organisations (European Space Agency) (Antropov et al. 2022). Much of today's forest area data are collected using laser scanning techniques and images taken from

aircraft and, more recently, drones. Satellites orbiting the Earth in predetermined orbits and photographing its surface in various bands of the electromagnetic spectrum are also used to create imagery (Pizarro et al. 2022). In addition to the visible light range, the near and mid-infrared, among others, are also captured. Using such data, it is possible to determine the various land cover parameters in the study area (Feng et al. 2022; Łaszewski et al. 2022). All recorded radiation is divided into several channels with different wavelengths. The data from the Landsat 8 satellite launched by NASA (National Aeronautics and Space Administration) on 11 February 2013 are divided into 11 channels, which facilitate its use in various analyses. In this study, channels 2–7 were used to delineate forest areas in the Polish part of the Carpathian Mountains. In order to automate the process of data analysis, the above-mentioned Google Earth Engine (GEE) platform was used, thanks to which, by writing a script in the JavaScript language, it is possible to analyse large amounts of data in a given area without worrying about the computing power of one's computer. In view of the above advantages, the aim of this study was to assess the usefulness of the GEE platform for delineating forest areas in the Polish Carpathians on the basis of Landsat satellite data. Using Earth Engine libraries, a script was written in an interactive environment, which covered the selection of appropriate data and their preparation, as well as supervised classification of images, assessment of its accuracy, calculation of forest area and graphical presentation.

Today, scientists carry out multitemporal and large-scale analyses of the state of the environment (Hixson et al. 1982). This is made possible, among other things, by data acquired from satellites (Wulder et al. 2019). One such analysis is the content classification of multispectral imagery to determine land cover classes. Land use maps have found wide application in monitoring environmental change, both of natural and anthropogenic origin (Ciołkosz and Białousz 2008). Among the many applications detailed in the literature are determining the extent of damage caused by a flood wave, updating soil maps, forecasting crop yields and assessing the condition of forests. The latter was studied by researchers at the Jagiellonian University in the FORECOM project, one of whose products was a map of forest distribution in the Polish Carpathians (Kozak and Kaim 2016). It was developed on the basis of a number of data

sets including Topographic Objects Database (BDOT) at a scale of 1:10 000, Numerical Forest Map (FNM) made available on the Forest Data Bank (FDB) website, as well as aerial and satellite images. The researchers encountered difficulties in integrating these data due to different definitions of forest. They also mention that the forest cover of municipalities calculated by them is higher than that reported by state statistics. Publications have also addressed the topic of mapping land cover maps over large areas based on satellite images. Such image classification is usually hampered by the lack of good training and validation data (Knorn et al. 2009).

Among environmental analysts, the GEE platform is becoming increasingly popular, as it is ultimately intended to monitor the state of the environment on a global scale. For the user, this means ease of processing large data sets, which is attractive from a time-saving perspective (Gorelick et al. 2017). Google has developed a number of algorithms, including the classification of the supervised content of satellite imagery, which was used in this study. However, before such a process can be carried out, the data must be properly prepared. No algorithm has been found on the platform in question for atmospheric correction of the imagery, i.e., that which results in reflection values at the Earth's surface. However, it appears that in many analyses, such a correction is omitted due to the lack of data to define a correct atmospheric model (Osińska-Skotak 2005). An object-based classification is also currently used, but this is not available in GEE.

METHODS

GEE is a platform for processing satellite imagery and other spatial data “in the cloud” (Google Developers 2020; Yang et al. 2022). The main use of this environment is to perform analysis and visualisation on a global scale, which often involves processing very large data sets. However, the performance of the personal computer is no longer an issue as the entire computational process takes place on Google's servers. All that is required is for the user to write an algorithm in one of the relevant programming languages (JavaScript or Python) using the GEE libraries. Not only satellite images, but also aerial photographs, numerical terrain models (DTMs), atmospheric, weather and climate data, as well

as data on land cover, its temperature and much more are made available. Users can also use their files, both vector and raster. To access the platform, a registration form needs to be completed, along with the purpose of use. Extensive documentation on how to work in GEE is provided online.

The platform consists of four components:

- spatial data sets (data sets)
- Google server computing infrastructure (computer power)
- API (application programming interface), in this case in JavaScript or Python
- Code Editor, a web-based integrated development environment designed to perform calculations quickly and transparently (no Python scripting here).

This paper uses the Code Editor, which is indicated for the novice user of the platform. In the central part of the window, there is a box for entering code. The left panel contains available methods and sample scripts, including user-created scripts. On the right is the output console, while at the bottom is an interactive map to which raster and vector layers can be added. The basic data structures in GEE are Images and Features, which correspond to raster and vector data models. Multiple objects of the same type can be combined to form Collections (Image/Feature Collection). There are also structures known from programming languages such as dictionaries, lists, arrays, numbers and character strings. Programming in GEE can be done by calling methods assigned to specific classes, ready-made algorithms or user-defined functions. On the other hand, displaying text, object metadata and calculation results on the console is done via the `print()` command. JavaScript is a scripting programming language whose most popular application is web pages. It is responsible for their interactivity, i.e., responding to user-induced events.

Some of the operations programmed by the user are performed in the browser, i.e., on the client side. On the other hand, only those related to Earth Engine library objects are executed on the server. These can be easily recognised in the collection as they start with the letters `ee`, for example, `ee.Image`. Combining these functions can cause unexpected errors. A good help in diagnosing why problems occur is to include a section called Profiler, which displays information about the various stages of the calculation, as well as the time in which they were performed. The most common error is incor-

rect code syntax, which is often picked up by the Editor with a solution provided.

The best-known source of satellite imagery is the Landsat programme run by the NASA and the United States Geological Survey (USGS). It was established in the 1970s, making it particularly useful for monitoring long-term changes in our planet's environment. Currently, the spatial resolution of the imagery is 30 metres. Landsat images are assigned to categories that reflect the quality of the data. The first of these is Real-Time data, which only goes into Tier 1 or Tier 2 after a certain period of time. The condition for being placed in a Tier 1 collection is a Root Mean Square Error value of less than 12 metres. Classification into Tier 2 may be due to significant image cloudiness. In this study, the image collection selected was: USGS Landsat 8 Collection 1 Tier 1 Raw Scenes, which covers the period from April 2013 to February 2019.

Remote sensing is the process of remotely acquiring information about the Earth's surface through aerial photographs and satellite imagery. Sensors used record electromagnetic radiation in certain ranges of the electromagnetic spectrum (Milewski 2010). Assuming that the spectral curves of individual objects are different, it is possible to identify them in the imagery. Classification of multispectral data involves their automatic interpretation for the purpose of mapping thematic maps, for example, land cover maps. There are two classification methods: supervised and unsupervised. The first requires the definition of training fields, i.e., land fragments considered to be representative of a given class. By statistically analysing the pixels in selected channels from the defined areas, the computer assigns each pixel to an appropriate category. For example, the following algorithms are appropriate here: maximum likelihood, cluster centre of gravity or decision tree (Mularz and Rutkowski 1995). In this work, the Classification and Regression Trees (CART) method was used, which is based on a binary recursive partitioning of a set of observations. The parent points of a decision tree are divided into two node points—descendant, that is, binary, while recursive involves treating the descendant node as the parent node further down the line (Łapczyński 2002).

The result of the classification should be viewed through the prism of an assessment of its accuracy, preferably performed on the basis of test fields. The basis for

determining accuracy is the error matrix, in which the rows correspond to the actual ground coverage and the columns to the classes into which the pixels have been classified by the classifier (Tab. 1). Each value is simply the number of pixels and it is denoted by the letter n . It is easy to see that the diagonal of the matrix contains pixels correctly classified. From this matrix, the total classification accuracy, the producer and user accuracies, and the kappa coefficient expressing the total classification error can be calculated. The first characteristic is the quotient of the sum of correctly classified pixels and their total number. Producer's accuracy is calculated as the ratio of the correctly classified pixels of a class to the total number of pixels in the actual class. In contrast, consumer's accuracy is expressed by the ratio of the number of pixels correctly classified of a given class to the total number of pixels of that class in the resulting image of the classification.

Table 1. Classification error matrix in Google Earth Engine

| | Class 0 | Class 1 | Class 2 | Class 3 |
|---------|----------|----------|----------|----------|
| Class 0 | n_{00} | n_{01} | n_{02} | n_{03} |
| Class 1 | n_{10} | n_{11} | n_{12} | n_{13} |
| Class 2 | n_{20} | n_{21} | n_{22} | n_{23} |
| Class 3 | n_{30} | n_{31} | n_{32} | n_{33} |

The error matrix is not only used to determine the overall accuracy characteristics but is also a concrete indication of how many pixels of a class have been classified as different. This is useful for incrementally improving classification by observing changes in the matrix. A programme was developed in the interactive environment of the Code Editor to carry out supervised classification of Landsat satellite imagery for the purpose of delineating forest areas in the Polish Carpathians. The diagram below shows the individual stages of the algorithm (Fig. 1).

The first stage involved loading the layer of the study region, selection of appropriate satellite images and vectorisation of fields considered representative for each class. In the next stage, the fields were divided into training and validation data and a supervised classification of the images was performed, along with an accuracy assessment. The third stage consisted of processing the classified image to determine the area of the forest class and the forest cover index. Finally, the resulting

layers were visualised in the GEE map window and the resulting raster was exported to Google Drive.

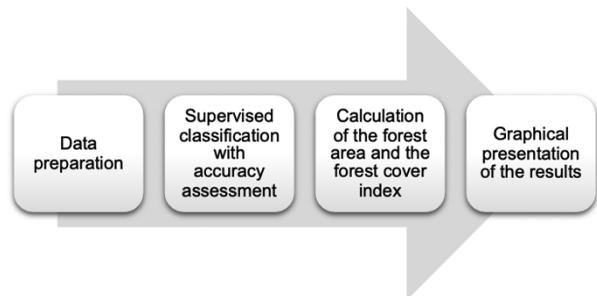


Figure 1. Schematic of the algorithm

RESULTS

A vector layer of the Polish Carpathian region in Esri Shapefile (.shp) format was loaded into the user's account as an Object Collection, which was then added to the script using its ID read in properties (Fig. 2).



Figure 2. Study area—Polish Carpathian region

The selected image collection had to be filtered to select images covering the study area and the least clouded. The region layer was given as the bounding area, while the months of June, July and August were used as the search period. The collection is accessed by entering its LANDSAT/LC08/C01/T1 ID stored in the Landsat data catalogue. The GEE library contains several algorithms related to Landsat imagery. One of these is Simple Composite, which creates a mosaic from the images by first converting the raw scenes to a Top of Atmosphere (TOA) reflectance value. In addition, a Cloud Score value is assigned for each pixel to indicate its degree of cloud cover, and the least clouded satellite scenes are later selected based on this. In the

application documentation, we find that the terms composition and mosaic in GEE are used interchangeably. In the present study, the above-mentioned algorithm was used for the prefiltered Karpaty image collection, and the mosaic was then cropped to regions using the clip() method (Fig. 3).

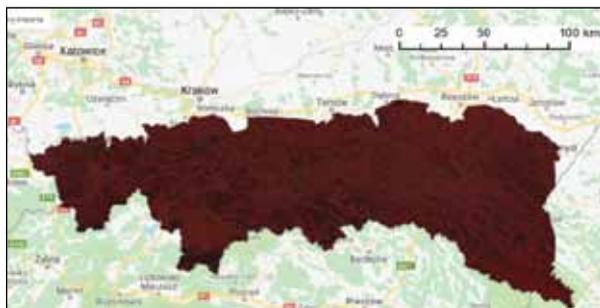


Figure 3. Colour composition from near infrared, green and blue channels

The basic activity to be carried out before supervised classification is the collection of data considered representative of the classes. Thanks to the ability to draw points, lines and polygons interactively in the map window, training fields (Fig. 4) were conveniently indicated. As the main objective was to delineate forest areas, the number of classes was narrowed down to 4—forest, water, built-up and other areas. When defining a new layer, it is necessary to set its type as Object Collection and the property that will be needed during subsequent classification.



Figure 4. Example of a drawn training field

The dialogue box for creating a forest class, where the value of the land cover property is 0, is shown in

Figure 5. The declarations of interactively drawn layers appear as code at the beginning of the script in a separate *Imports* section.

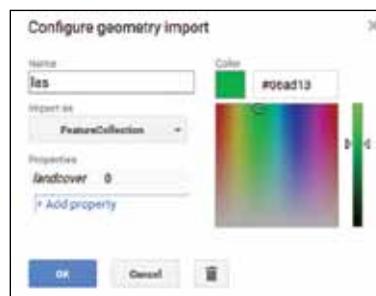


Figure 5. Dialogue box for adding a new layer

The data prepared in this way were merged into a single Object Collection using the merge() method. A list of specific channels (B2, B3, B4, B5, B6, B7), based on which the classification will be performed, was defined. Using the sampleRegions() method, a sample of image pixels inside the previously drawn regions was indicated. The parameters given are as follows: the Object Collection, which is the layer with the training fields, and a list of all their properties, in this case only the land cover. When working on rasters in GEE, there are many times a need to define the scale, i.e., the pixel resolution of the resulting image. Here and throughout the algorithm, this was set as 30 metres. It was decided that the collected data would be split into training and test data in a ratio of approximately 70/30, with the former being used to train the classifier and the latter to assess the accuracy of the results. For the Object Collection named data, a column was created and populated with pseudo-random values in the range 0.0–1.0. It was then filtered using lt(), which means less than, and gte(), which means greater than or equal. This resulted in two sets of pixels: a training set, representing about 70% of the total data, and a test set, the remaining 30%. First, an empty CART classifier was created and trained on the training object collection. The classProperty parameter is the previously mentioned land cover property storing the class numbers. In the last line of the following code, the classification of the satellite imagery is performed based on the predefined channel list named bands and the classifier. The resulting image is stored in the classified variable.

In GEE, it is possible to assess the accuracy of the classification results from both training and validation data. The former is done using the `confusionMatrix()` method, which creates a matrix in which the rows correspond to the actual land cover and the columns to the classes into which they were classified by the classifier. Total classification accuracy, producer and user accuracy, and the kappa coefficient were calculated. In the script written, these values are not assigned to variables, but are immediately displayed on the console via the `print()` command. The trained classifier was used to classify a test data set representing approximately 30% of the total data sample. This determined validation accuracy, which is more reliable than training accuracy.

From the classified image, a binary image `las_bin` was created, whose pixels took values equal to 1 for the forest class or equal to 0 for the other classes (Figure 6). The next line of code uses the `ee.Image.pixelArea()` method, which creates an image for which the value of each pixel is equal to its area expressed in square metres. Dividing these values by 10 000 converts them into hectares. The final image needed to calculate the area is created by multiplying the previous two `area_ha` and `forest_bin`.

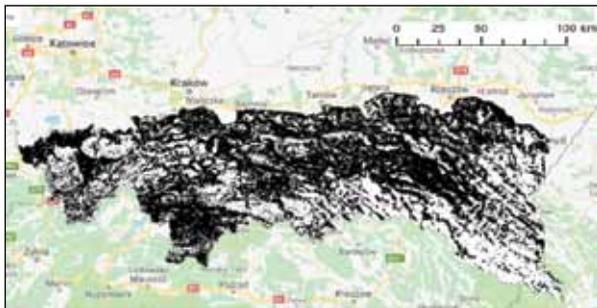


Figure 6. Binary image (white represents forests)

The forest area (variable `pow`) was calculated using the `reduceRegion()` method, which aggregates data within a defined region. In this case, the statistics of the sum of pixel values were used. The `maxPixels` parameter set as `1e9` means that the largest possible converted number of pixels is 1,000,000,000, while `bestEffort` taken as `false` protects the function from scaling up if it is difficult to cope with a large data set. Otherwise, it would not be entirely clear at what spatial resolution the calculation was performed. The area of the entire region

was calculated in the same way. By calculating the percentage ratio of the forest area to the area of the whole area, the forest cover index of the Polish Carpathians was obtained. The determined values were displayed on the console. The Earth Engine library methods `divide()` and `multiply()` should be used here, rather than the usual arithmetic operators.

Earth Engine objects can be displayed in the map window on one of the available map primers. This is done via the `Map.addLayer()` method, for which the following are given as arguments: the object name, the visualisation parameters (in curly brackets: `{}`) and the layer label to be displayed in the legend. The visualisation options vary in compositions in their specific channels, while for other objects they can be transparency or a colour palette.

In this algorithm, such a palette has been predefined by providing Cascading Style Sheets (CSS) colour names for all classes. Figure 7 shows the visualisation in the map window of the classified satellite image. The entire algorithm is included in this paper as an appendix, together with comments on the various calculation steps for better code clarity.



Figure 7. Classified image in the Google Earth Engine map window

However, if such visualisation in the map window is not sufficient or there is a need for further analysis outside the GEE environment, the resulting image can be exported by adding code at the end of the programme with an algorithm that will place the export task in the `Tasks` table. In this case, the process of preparing the image for export took a few minutes. At the end, in the displayed dialogue box, it is possible to modify predefined parameters such as the spatial resolution of the raster and to choose from the available places to save the file. The saved raster was processed in the QGIS software, and a map composition was made in order to

better present the designated land cover of the Polish Carpathians.

Considering the available sources of information on forests, the Bureau for Forest Management and Geodesy was asked to provide a layer of natural forest lands. The obtained boundaries of the Polish part of the Carpathian Mountains were loaded into the GEE platform in order to run the algorithm. The calculated forest area was 910 149 hectares, with the total area of the Carpathian region equal to 1 939 010 hectares. Consequently, the forest cover index of the Polish Carpathians was determined as 46.9%. It is worth recalling that the forest area was calculated as a total pixel area, therefore with a 30-metre spatial resolution.

Table 2. Error matrix for validation data

| | Forest | Water | Other area | Built-up area |
|---------------|--------|--------|------------|---------------|
| Forest | 13 302 | 0 | 313 | 5 |
| Water | 6 | 13 247 | 45 | 45 |
| Other areas | 541 | 14 | 7599 | 90 |
| Built-up area | 1 | 10 | 143 | 723 |

To be able to assess the reliability of these results, accuracy analyses were carried out on both training and validation data. Considering the latter to be more reliable, Table 2 contains the error matrix for the validation data, while Table 3 contains the calculated accuracies.

Table 3. Results of the accuracy analysis

| Class | Manufacturer accuracy | User accuracy |
|------------------|-----------------------|---------------|
| Forest | 0.976 | 0.960 |
| Water | 0.993 | 0.998 |
| Other | 0.922 | 0.938 |
| Construction | 0.824 | 0.837 |
| Overall accuracy | 0.966 | |
| Kappa | 0.950 | |

Referring to the entire classification, 97% of the pixels comprising all test fields were assigned to the correct classes. The laser class manufacturer's accuracy value of 0.976 means that more than 97% of the pixels were correctly classified. The lowest accuracy was obtained for buildings. When visually inspecting the classified image, it turned out that the higher parts of

the mountains, covered with snow all year round, were considered by the algorithm to be built-up areas. In such a situation, a separate class for these areas would have to be distinguished to prevent this in the delimitation of built-up areas, but this is of little importance in the delimitation of forest complexes. While working on the algorithm, errors were noticed in the area of the Tatra Mountains, where the vegetation of the dwarf pine layer was classified as forest. These were eliminated by creating additional training fields, which were categorised as other. The calculated accuracy values may be overestimated, which is due to the fact that the sample of validation pixels is part of the set indicated by the author on the basis of visual interpretation. Consequently, these are fragments of terrain that are easy to recognise and therefore also simpler for the classifier to classify. The above accuracy results can be considered good; however, in order to properly assess the correctness of the classification, a check would have to be made on several smaller areas of varying terrain. Such a check should be based on other studies, for example, higher resolution aerial photographs or field interview.

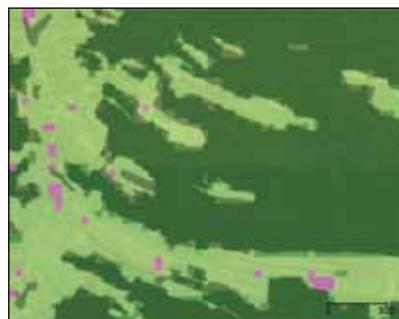


Figure 8. Comparison of the classified image with the forest mask of the FORECOM project

Figure 8 shows a comparison of the classified image with the forest areas delineated from the FORECOM project (validity 2010–2015), parts of which were made available to the authors by researchers of the Institute of Geography and Spatial Management of the Jagiellonian University. Dark green indicates forests, pink indicates urbanised areas and light green indicates other areas. A vector layer of the extracted forest mask was superimposed on the classified image. The image shows the area around Gliczarowo Górne, i.e., the Podhale region. Due to the dispersed nature of the spatial distribution of

streets and buildings, urbanised areas were depicted as very small groups of pixels.

When looking for information on forests growing in the Polish part of the Carpathians, it is worth noting that different sources assume different boundaries of this area. For example, in the FORECOM project's user manual, one can find information that the forest cover in the Polish Carpathians today is approximately 47% and is calculated on an area consisting of 194 municipalities recognised by the Carpathian Convention as being entirely within the Carpathians (Kozak and Kaim 2016). However, the FORECOM project's forest map represents land use, not land cover. The latter is determined in this study from satellite imagery. Differences between land use and land cover can occur in deforested areas or forest succession. Table 4 shows the results of calculations of the developed algorithm on the GEE platform, as well as data read or calculated from other sources.

Table 4. Results comparison with other sources

| Source | Forest area [ha] | Carpathian area [ha] | Forest cover coefficient [%] | Topicality |
|---------------------|------------------|----------------------|------------------------------|------------|
| Google Earth Engine | 910 149 | 1 939 010 | 46.9 | 2013 |
| Forest Data Bank | 772 884 | 1 937 602 | 39.9 | 2013 |
| Corine Land Cover | 934 494 | 1 937 602 | 48.2 | 2012 |

The FDB is an information system used, among other things, to collect and make available information on forests. Tabulated data on the area of forests on forest land by form of ownership in the order of natural and forest lands were downloaded from www.bdl.lasy.gov.pl. The report shows that as of 1 January 2013, the area of forests on forest land was 772 884 hectares, while the forest cover was 39.9% for the total area of the Polish part of the Carpathians equal to 1 937 602 hectares. In this case, a minimum area of 0.1 hectare was assumed as forest, according to Article 3 of the Forest Act (Kancelaria Sejmu 2019).

Another data source is the European project Corine Land Cover 2012, which aims to provide regular information on land cover in Europe. The database was created through computer-aided visual interpretation of

satellite images with a geometric resolution of 25 metres but note that areas with a minimum size of 25 hectares were selected. Of all land cover forms, the class Forests was selected, i.e., areas with dense cover and a crown height of more than 5 metres. By calculating the statistics for this class, the area of forests in the Polish Carpathians was determined to be 934 494 hectares and the forest cover ratio to be 48.2%.

The area of one pixel of the Landsat satellite image is 0.09 hectare, so this is the smallest area that can be assigned to the forest class by the developed algorithm. It is much smaller than those separated in the above studies. The forest cover ratio calculated in GEE differs by 1.3% from that determined from the Corine Land Cover project data and by 7% from the value given in the FDB compilation. In order to improve the classification results, more attention should be paid to the selection of training fields. The above values are only indicative, due to the different ways in which forest areas are delineated.

When assessing the usefulness of the GEE platform for delineating forest areas in the Polish Carpathians on the basis of Landsat satellite data, several factors should be taken into account. The amount of effort involved in the preparation of the algorithm compared with the studies mentioned in this paper was much less. The undoubted advantage is that the calculations are performed on the server side and there is no need to download large volumes of data to your computer. This makes it possible to carry out analyses on low-performance hardware and on different operating systems, as only a well-functioning web browser is required. Furthermore, the server does not have to wait long for a response; in the case of the algorithm under development, all calculations in the Google Chrome browser took just two minutes. The platform's high performance is due to the parallelisation of the calculations, which involves splitting the operations so that they are performed in parallel on the server. This process is automatic and does not affect the results but allows them to be obtained quickly. The Code Editor is an interactive environment that allows the user to: add geometric objects by drawing them in the map window, display the calculation results in the map window, and, using the Inspector tool, read the properties of a given image pixel or vector layer. For supervised classification, interactivity improves the indication of training fields. This is a convenient method, as it is possible to use a map

base or another vector layer that has been loaded. To transfer such points, lines or polygons to another script, it is sufficient to generate and then copy the corresponding JavaScript code, which is useful when working with multiple algorithms. The user has more than a dozen supervised and several unsupervised classification methods available. The usability of the platform for those with little programming knowledge is an undoubted advantage. Extensive documentation is available on the Internet to introduce the beginner to script writing, as well as to show advanced uses of finished applications. Of interest from the point of view of long-term script development may be the version control system, which keeps an eye on changes to the code and also allows any previous version to be restored. What stands out about GEE is the ease of finding satellite images of interest. Everything is done by filtering the selected Image Collection both spatially and temporally. The `sort()` method allows sorting according to the information contained in the metadata, for example, in order from the least cloudy satellite scenes to the most. The quick selection of source data significantly reduces working time by eliminating the need for traditional downloading to one's own hard drive from portals providing these data. The user can immediately view the retrieved images. A map overlay in the map window facilitates orientation when defining training fields, filtering images and visually analysing classification results. The platform provides a high degree of freedom to manipulate the display parameters of objects and images.

DISCUSSION

New methods of working with data are constantly being developed and research processes are constantly being improved. One example is the GEE platform discussed above, where it only takes a few minutes to carry out an analysis on a regional, national or continental scale. The timeliness of data and the diversity of sources are satisfactory. The time taken to obtain information is significantly reduced, but at the expense of constant learning to use new tools. The GEE environment is constantly being developed and has great potential already. The author programmed the process of delineating forest areas on the example of the naturally valuable area of the Polish Carpathians using GEE algorithms. The procedure

included preprocessing of satellite images, supervised classification with accuracy assessment and determination of the forest cover index of the Polish Carpathians. The written algorithm was used to characterise the specifics of working in the platform's Code Editor and to assess its suitability for this type of analysis. Among the arguments confirming the validity of the use of the platform are the speed of calculations due to their parallelism on the server, as well as the interactivity of the environment facilitating the indication of training fields and ongoing analysis of the results while working on the code. Satellite scenes once made available as individual files of large size can be used without downloading to disk, and collecting them into Image Collections has enabled rapid selection. A great deal of freedom to test different computational options is provided by a dozen supervised and several unsupervised classification methods. The use of the platform allowed good quality results to be obtained in a relatively short time, as confirmed by comparing the calculations with similar analyses carried out by other researchers. It was concluded that the GEE platform could be an effective tool for delineating forest areas in the Polish Carpathians. Given the advancement of satellite technology, better and better data quality can be expected, which opens up great opportunities for this type of research in GEE in the future.

ACKNOWLEDGEMENTS

This work was funded by AGH University of Science and Technology research subvention no. 16.16.150.545.

REFERENCES

- Antropov, O. et al. 2022. Intercomparison of Earth Observation Data and Methods for Forest Mapping in the Context of Forest Carbon Monitoring. In: IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium. IEEE, 5777–5780.
- Ciołkosz, A., Białousz, S. 2008. Zastosowanie teledetekcji satelitarnej w badaniach środowiska w Polsce. *Nauka*, 3, 79–96.
- Feng, S. et al 2022. Land Use/Land Cover Mapping Based on GEE for the Monitoring of Changes in

- Ecosystem Types in the Upper Yellow River Basin over the Tibetan Plateau. *Remote Sensing*, 14, 1–25. DOI: 10.3390/rs14215361
- Google Developers. 2020. Google Earth Engine. Available at <https://developers.google.com/earth-engine> (access on 25 February 2023).
- Gorelick, N. et al. 2017. Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 202, 18–27. DOI: 10.1016/j.rse.2017.06.031
- Hixson, M.M., Bauer, M.E., Scholz, D.K. 1982. An assessment of Landsat data acquisition history on identification and area estimation of corn and soybeans. *Remote Sensing of Environment*, 12, 123–128. DOI: 10.1016/0034-4257(82)90032-3
- Kancelaria Sejmu. 2019. Ustawa o lasach. *Dziennik Ustaw*, 1–53.
- Knorn, J. et al. 2009. Land cover mapping of large areas using chain classification of neighboring Landsat satellite images. *Remote Sensing of Environment*, 113, 957–964. DOI: 10.1016/j.rse.2009.01.010
- Kozak, J., Kaim, D. 2016. Forecom: podręcznik użytkownika. IGiGP UJ, Kraków.
- Kozioł, K., Maciuk, K. 2020. New heights of the highest peaks of Polish mountain ranges. *Remote Sensing*, 12, 1446. DOI: 10.3390/rs12091446
- Łapczyński, M. 2002. Badania segmentów rynku motoryzacyjnego z zastosowaniem drzew klasyfikacyjnych (CART). *Zeszyty Naukowe Akademii Ekonomicznej w Krakowie*, 586, 88–102.
- Łaszewski, M., Fedorczyk, M., Stępniewski, K. 2022. The Impact of Land Cover on Selected Water Quality Parameters in Polish Lowland Streams during the Non-Vegetative Period. *Water*, 14, 3295. DOI: 10.3390/w14203295
- Maciuk, K. et al. 2021. Altitude on Cartographic Materials and Its Correction According to New Measurement Techniques. *Remote Sensing*, 13, 444. DOI: 10.3390/rs13030444
- Milewski, W. 2010. Geomatyka w Lasach Państwowych. Część 1. Podstawy. CILP, Warszawa.
- Mularz, S., Rutkowski, J. 1995. Zastosowanie teledetekcji w ochronie środowiska. Ochrona środowiska przyrodniczego i zasobów mineralnych. Materiały studium podyplomowego AGH w Krakowie. CPPGSMiE, PAN, Kraków, 93–106.
- Osińska-Skotak, K. 2005. Wpływ korekcji atmosferycznej zdjęć satelitarnych na wyniki cyfrowej klasyfikacji wielospektralnej. *Acta Scientiarum Polonorum. Geodesia et Descriptio Terrarum*, 4 (1), 41–53.
- Pizarro, S.E. et al. 2022. Mapping Land Cover Types for Highland Andean Ecosystems in Peru Using Google Earth Engine. *Remote Sensing*, 14, 1562. DOI: 10.3390/rs14071562
- Szramka, H., Adamowicz, K. 2020. Forest development and conservation policy in Poland. *Folia Forestalia Polonica, Series A – Forestry*, 62 (1), 31–38. DOI: 10.2478/ffp-2020-0004
- Wulder, M.A. et al. 2019. Current status of Landsat program, science, and applications. *Remote Sensing of Environment*, 225, 127–147. DOI: 10.1016/j.rse.2019.02.015
- Yang, L. et al. 2022. Google Earth Engine and Artificial Intelligence (AI): A Comprehensive Review. *Remote Sensing*, 14, 3253. DOI: 10.3390/rs14143253